Examiners Amendment

1. An examiner’s amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner’s amendment was given in a telephone interview with *Robert P. Lord regsiteration 469,479*, on September 23, 2008. Claims are being amended to overcome potential 101 issues and clarify claim limitations.

The application has been amended as follows:

IN THE SPECIFICATIONS

Please amend paragraph [0048] as follows:

[0048] In addition, embodiments of the present invention further relate to computer storage products with a computer-readable medium that have computer code thereon for performing various computer-implemented operations. The media and computer code may be those specially designed and constructed for the purposes of the present invention, or they may be of the kind well known and available to those having skill in the computer software arts. Examples of computer-readable media include, but are not limited to: magnetic media such as hard disks, floppy disks, and magnetic tape; optical media such as CD-ROMs and holographic devices; magneto-optical media such as floptical disks; and hardware devices that are specially configured to store and execute program code, such as application-specific integrated circuits (ASICs), programmable logic devices (PLDs) and ROM and RAM devices. ~~Additionally, such computer-readable media include, but are not limited to, carrier waves transmitted through wires or optical transmission media (e.g., fiber optic cable) or other transmission media.~~ Examples of computer code include machine code, such as produced by a compiler, and files containing higher level code that are executed by a computer using an interpreter.

IN THE CLAIMS

2. Please amend claims 1, 9, 18 and cancel claims 3, 4, 10, 12, 13, 19, 21 and 22 as attached herein see pages 6 - 14.

See attached document as proposed by Applicant to amend claims

Reasons for Allowance

3. Examiner has reviewed and considered Applicant's arguments and comments per Applicant's response of 05/21/08, and based on Applicant's arguments on pages 2 – 4 as well current examiners amendments as presented below regarding claims 1, 2 ,5 – 9,11, 14 – 18, 20 and 23 – 26 all remaining claims are now in condition for allowance.

The following is an Examiner's statement of reasons for allowance.

The prior art of record does not teach or fairly suggest at least the limitations of:

"…wherein the relation comprises a matrix of argument signatures that relate the sets of dummy arguments with associated specific functions for the generic function, and wherein the generic function call comprises at least one argument and the first sorting through the relation comprises sorting and comparing each dummy argument of an argument signature with the least one argument of the generic function call until a parameter mismatch is determined for the dummy argument at which time the first sorting skips to a next argument signature and continues sorting and comparing the at least one argument of the generic function call with the dummy arguments of the next argument signature…" and as best illustrated by figs. 1b and 2, in such a manner as recited in independent claims 1, 9 and 18, as indicated in Applicant's response on pages 2 – 4.

Therefore, all remaining 1, 2, 5 – 9, 11, 14 – 18, 20 and 23 – 26 are in condition for allowance.

> Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Correspondence Information

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chuck Kendall whose telephone number is 571-2723698. The examiner can normally be reached on 10:00 am - 6:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on 571-2723695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Chuck O Kendall/
Primary Examiner, Art Unit 2192

PROPOSED CLAIM AMENDMENTS

1. (Currently Amended) A method for detecting and analyzing errors in a generic function call comprising:

providing a generic function call;

providing a relation that includes sets of dummy arguments associated with the generic function call;

first sorting through the relation to determine whether the generic function call contains errors;

if the first sorting determines that the generic function call includes an error, second sorting through the relation to determine a failure mode for the generic function call; and

providing an error message that identifies that the generic function call has an error and provides information about the failure mode of the generic function call as determined by the second sorting,

wherein the relation comprises a matrix of argument signatures that relate the sets of dummy arguments with associated specific functions for the generic function, and

wherein the generic function call comprises at least one argument and the first sorting through the relation comprises sorting and comparing each dummy argument of an argument signature with the least one argument of the generic function call until a parameter mismatch is determined for the dummy argument at which time the first sorting skips to a next argument signature and continues sorting and comparing the at least one argument of the generic function call with the dummy arguments of the next argument signature.

2. (Original) The method of Claim 1 wherein the relation comprises a generic function definition table.

3. (Cancelled)

4. (Cancelled)

5. (Currently Amended) The method of Claim [[3]] 1 wherein the second sorting through the relation to determine the failure mode of the generic function call comprises:

    identifying which parameters of the generic function call contain errors; and wherein providing an error message comprises providing an error message that identifies which parameters contain errors in the generic function call.

6. (Original) The method of Claim 5 wherein said second sorting further comprises clarifying the nature of the error to determine a type of error present in the parameter that contains errors; and wherein providing an error message comprises providing an error message that identifies which parameter contains the error and the type of error.

7. (Original) The method of Claim 6 wherein clarifying the nature of the error to determine a type of error present in the parameter includes determining that the type of error is at least one of a rank error, a type error, and a kind error.

8. (Currently Amended) The method of Claim [[3]] 1 wherein the second sorting determines which argument signature has the most correct format matches with the generic function call; and

wherein a mis-matched parameter in the argument signature having the most correct format matches is identified as a parameter containing an error.

9. (Currently Amended) A method for detecting and analyzing errors in a generic function call constructed in accordance with a programming language protocol, the method comprising:

providing a generic function call that can invoke a plurality of different specific functions which can be varied by selecting different parameters of the generic function call;

determining whether the generic function call contains errors; and

providing an error message that includes information about the failure mode that caused the errors in the generic function call,

wherein determining whether the generic function call contains errors comprises:

providing a relation that defines the generic function call in terms of sets of dummy arguments comprising argument signatures representative of specific functions defined for the generic function call,

first sorting through the relation to compare the generic function call with the dummy arguments of the relation,

where the generic function call correctly matches a format for one of the arguments signatures of the relation, the generic function call is deemed to have no error,

where the generic function call does not correctly match a format for any of the arguments signatures, the generic function call is deemed to have an error,

wherein identifying the nature of the errors comprises:

recognizing that the first sorting has determined that an error is present in the generic function call;

second sorting through the relation to determine if the failure mode in the generic function call can be further clarified, and

wherein providing an error message comprises providing an error message that includes information about the failure mode determined by the second sorting,

wherein the relation comprises a matrix of argument signatures that relate the sets of dummy arguments with associated specific functions for the generic function, and

wherein the generic function call comprises at least one argument and the first sorting through the relation comprises sorting and comparing each dummy argument of an argument signature with the least one argument of the generic function call until a parameter mismatch is determined for the dummy argument at which time the first sorting skips to a next argument signature and continues sorting and comparing the at least one argument of the generic function call with the dummy arguments of the next argument signature.

10. (Cancelled)

11. (Currently Amended) The method of Claim [[10]] 9 wherein the relation comprises a generic function definition table.

12. (Cancelled)

13. (Cancelled)

14. (Currently Amended) The method of Claim [[10]] 9 wherein the second sorting through the relation to determine the failure mode of the generic function call comprises:

identifying which parameters of the generic function call contain errors; and

wherein providing an error message comprises providing an error message that identifies which parameters contain errors in the generic function call.

15. (Original) The method of Claim 14 wherein said second sorting further comprises clarifying the nature of the error to determine a type of error present in the parameter that contains errors; and wherein providing an error message comprises providing an error message that identifies which parameter contains the error and the type of error.

16. (Original) The method of Claim 15 wherein clarifying the nature of the error to determine a type of error present in the parameter includes determining that the type of error is at least one of a rank error, a type error, and a kind error.

17. (Currently Amended) The method of Claim [[10]] 9 wherein the second sorting determines which argument signature has the most correct format matches with the generic function call; and wherein a mismatched parameter in the argument signature having the most correct format matches is identified as a parameter containing an error.

18. (Currently Amended) A computer readable media including computer program code for detecting and analyzing errors in a generic function call, computer readable media comprising:

computer code for providing a generic function call that can invoke a plurality of different specific functions which can be varied by selecting different parameters of the generic function call;

computer code for determining whether the generic function call contains an error;

computer code for identifying the nature of the error; and

computer code for providing an error message that includes information about the failure mode that caused the error in the generic function call,

wherein the code for determining whether the generic function call contains errors comprises:

computer code for providing a relation that defines the generic function call in terms of sets of dummy arguments configured as arguments signatures representative of specific functions defined for the generic function call;

computer code for first sorting through the relation to compare the generic function call with the dummy arguments of the relation;

where the generic function call correctly matches a format for one of the argument signatures, the generic function call is deemed to have no error;

where the generic function call fails to correctly match a format for any of the argument signatures, the generic function call is deemed to have an error;

wherein the computer code for identifying the nature of the errors comprises:

computer code for recognizing that the first sorting has determined that an error is present in the generic function call;

computer code for second sorting through the relation to determine if the failure mode in the generic function call can be further clarified; and

wherein the computer code for providing an error message includes code for providing an error message that includes information about the failure mode determined by the second sorting,

wherein the relation comprises a computer readable matrix of arguments signatures that relate sets of dummy arguments with associated specific functions for the generic function call,

wherein the computer code for first sorting through the relation comprises computer code for sorting and comparing the generic function call with the dummy arguments of each argument signature until a parameter mismatch is determined for a dummy argument of the signature at which time the computer code for first sorting skips to a next argument signature and continues sorting and comparing the generic function call with the next arguments signature.

19. (Cancelled)

20. (Currently Amended) The computer readable media of Claim [[19]] 18 wherein the relation comprises a computer readable generic function definition table for the generic function call.

21. (Cancelled)

22. (Cancelled)

23. (Currently Amended) The computer readable media of Claim [[19]] 18 wherein the computer code for second sorting through the relation to determine the failure mode of the generic function call comprises computer code for identifying which parameters of the generic function call contain errors; and wherein the computer code for providing an error message comprises computer code for providing an error message that identifies which parameters contain errors in the generic function call.

24. (Original) The computer readable media of Claim 23 wherein said computer code for second sorting further comprises computer code for clarifying the nature of the error to determine a type of error present in the parameter that contains errors; and

wherein computer code for providing an error message comprises computer code for providing an error message that identifies which parameter contains the error and the type of error.

25. (Original) The computer readable media of Claim 24 wherein the computer code for clarifying the nature of the error to determine a type of error present in the parameter includes computer code for determining that the type of error is at least one of a rank error, a type error, and a kind error.

26. (Currently Amended) The computer readable media of Claim [[19]] 18 wherein the computer code for second sorting includes computer code for determining which argument signature has the most correct format matches with the generic function call;

wherein the computer readable media includes computer code for identifying the argument signature having the fewest mis-matched parameters when compared with the generic function call;

wherein computer readable media includes computer readable code for identifying which parameters are mis-matched and identifying those mis-matched parameters as the parameters containing errors; and

wherein the computer code for providing an error message comprises computer code for providing an error message that identifies the mis-matched parameter as a parameter that contains an error.

/Chuck O Kendall/

Primary Examiner, Art Unit 2192